

REMARKS

Claims 1-29 are currently pending in the subject application and are presently under consideration. New claim 30 has been added and is fully supported by the specification as filed. No new matter has been introduced.

Favorable reconsideration of the subject patent application is respectfully requested in view of the comments and amendments herein.

I. Rejection of Claims 1-29 Under 35 U.S.C. §103(a)

Claims 1-29 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Beaumont et al. (2004/0073511) (hereinafter “Beaumont”). Claims 1-29 stand rejected under 35 U.S.C. §102(e) as being anticipated by Beaumont *et al.* (2004/0073511) in light of Burke (US5423041) (hereinafter “Burke”). Applicants’ representative respectfully disagrees with the Examiner’s position for at least the following reasons.

Applicants’ claimed invention relates generally to user friendly modeling of complex business rule(s) in a computer system. In particular, the present invention relates to a system and method for tree-based rule composition *with natural language inline editing*. For example, the system can combine a familiar hierarchical structure, such as a computer tree, and inline editing of natural language based logical expression(s) to provide a general purpose logical expression builder. The term *natural language* is defined as “A language spoken or written by humans, as opposed to a programming language of a machine language”, (*see* Microsoft Computer Dictionary, 5th Ed.) The connotation of natural language inline editing thus includes editing of rules using plain human speech elements as compared to more typical systems that could include learning a programming language to facilitate editing in that specific programmatic language.

Generally, Beaumont relates to systems and methods that permit credit card processing organizations to apply user friendly entry of rules into a user interface to design and administer credit card programs. Accordingly, Beaumont discloses a method of providing bankcard services that include receiving at a host computer system a selection of a business rule from a library of business rules. The business rules may include information defining an event, a condition relating to at least one element associated with bankcard accounts, and information defining an action to be taken based on the condition upon the occurrence of the event. However, as is conceded in this and earlier Office Actions, Beaumont fails to disclose or

anticipate at least the novel aspect of employing *natural language inline editing*, as presently claimed. The Examiner turns to Burke to cure these deficiencies, however, Burke does not cure these deficiencies.

With regard to Burke, the reference generally relates to an Application Program Interface (API) for structuring a programming language based in part on rule-based programming. Burke states that rule-based programming is “another approach to *programming*” (*see* col. 1, ln. 22, emphasis added) in contrast to object-oriented programming and function-based programming. Thus, Burke is clearly discussing rule-based programming as a *general programming architecture*. To facilitate this general programming architecture, Burke presents a basic programming interface definition and states, “It would be advantageous if programmers could develop a problem domain ...and then develop rules to inference that domain...the rules could be directly coupled to an object-oriented application program, and the need to modify the application program would be minimal.” (col. 1, ln. 44-50). Burke continues to describe writing rule-based code and then compiling it for execution, “the **programmer writes an application...then codes the rules**...also codes a special header...inserts a rule calling sequence in the...program, which defines an inference class, **binds an inference class object to a rule**, adds user-defined class instances..., and calls an inference process...the **rules are compiled with a rules compiler...that code is then compiled and linked into the application program.**” (col. 1, ln 55 to col. 2, ln 5, emphasis added). Burke is discussing a relatively low level of programming for a computer to execute applications written in the rules-based language. As is well understood in the art, low level programming generally requires expertise in the programming environment and language such that low level programming is not typically considered “user friendly”. Thus, the programming events in Burke would be inappropriate for the user friendly natural language inline editing of business rules disclosed in the subject application.

Burke cannot be considered analogous art and thus obviousness *per se* has not been established.

Burke is in sharp contrast to the invention of Beaumont. Where Beaumont describes a user interface germane to unsophisticated users developing a credit card program, *Beaumont in no way implies that one of skill in low level rule-based programming is needed to set up a credit*

card offer. Similarly, the development of rules in the instant application are also oriented to users and a user friendly environment *not necessitating some mastery of low-level programming techniques such as compiling, linking, and writing code in the native tongue of a low level API*. This can be evidenced in both Beaumont and the instant application by examining the user interfaces presented in each and the corresponding descriptions in the specifications, in which it is clear that the input for rule making in Beaumont is though a user friendly interface and that there are numerous user interface options (*e.g.*, apply buttons, cancel buttons, help sections, discussion of using a mouse as a prominent means of interacting with the UI, ...) that are well understood by one of skill in the programming arts not to be associated with low level programming API's. This is in sharp contrast to the low level programming environment of Burke. Thus, the mere fact that Burke discusses rules and programming of rules for a computer is **not sufficient to rise to the level of making Burke analogous art** considering the vast difference between the types of programming described therein.

Burke cannot be combined with Beaumont to result in the disclosed subject matter without significant changes to Beaumont and thus obviousness *per se* has not been established.

Burke clearly describes their invention as needing to be programmed and compiled, as disclosed *supra*. In contrast, *Beaumont discloses forming rules by entering arguments, which can hardly be considered "programming", into a user interface, which can certainly not be considered "compiling"*. Further, Burke discloses **needing custom headers for the rules** (col. 8, ln. 61, not germane to the invention of Beaumont), **the presence of an inference engine** (col. 4, ln. 53-57, among others, also not germane to Beaumont), and that **the syntax of the rule is much like that of a C++ function** (*see* col. 5, ln. 3-4, **followed by a sample C++ style rule** at ln. 8-22, clearly not germane to Beaumont and certainly not representative of natural language editing). Thus, in order to combine the teachings of Burke into the teachings of Beaumont, the basic invention of Beaumont would necessarily be radically changed to accommodate the teachings of Burke. This is not permissible in a 35 U.S.C. § 103(a) rejection, and thus the rejection is improper as the Examiner has not presented a case of obviousness *per se*, where there is impermissible modification of the primary reference to combine the teachings.

Even where Burke *arguendo* presents a proper obviousness *per se* rejection, Burke fails to correct the deficiencies of Beaumont.

As stated *supra*, Burke teaches a rule-based programming interface and does not teach “*creation and modification of the rule based... upon natural language inline editing*”, as claimed. The inline editing of natural language based logical expression(s) of the claimed subject matter can be combined with hierarchical tree structures to provide a general purpose logical expression builder. **Beaumont fails to disclose or anticipate any such natural language inline editing.** Beaumont merely discloses that clients can select pre-designed business rules, modify rules in the library, or create their own rules. Burke does not cure this defect. **Burke merely teaches that rules-based programming can be combined with object-oriented programming to present an API for constructing a piece of software that is rule-based.** This is not *natural language inline editing*. Burke entails neither use of “natural language” nor “inline editing”. In contrast to the instant application, Burke presents editing in a programming language with syntax much like C or C++. The C-programming language and the C++ programming language are not “natural languages” (*see* definition for natural language provided *supra*) as would be understood by one of ordinary skill in the art, and moreover, the editing is at the function writing level prior to compiling executable code and not “inline”. Therefore, Burke does not teach natural language inline editing and thus *does not correct the deficiencies of Beaumont*. Thus, Burke, either alone or in combination with Beaumont, neither explicitly nor implicitly teaches every aspect of the claimed invention.

With regard to independent claim 1, the claim in part, recites a system facilitating rule composition comprising: *a rule composer that in response to a user input facilitates at least one of creation and modification of a rule, the rule composer further receiving information associated with the at least one of creation and modification of the rule based, at least in part, upon natural language inline editing.* Similar language is present in independent claims 22-24 and 25-26. This aspect is not obviated by Beaumont or Burke, either alone or in combination. Based on the foregoing, Beaumont cannot be properly combined with Burke to show obviousness as they are not analogous art. Further, the two references cannot properly be combined without substantial modification and thus also fail to obviate the present claims. Additionally, even were the cited art combined, each and every element of the claimed subject matter is not disclosed as recited in independent claims 1, 22, 23, 24, 25, and 26. Where claims

2-21 depend from independent claim 1 and claims 27-29 depend from independent claim 26, these dependent claims are also believed to be allowable. Therefore, it is respectfully requested that the rejection of claims 1-29 under 35 U.S.C. § 103(a) over Beaumont in view of Burke e withdrawn and the application passed to allowance at an early date.

CONCLUSION

The present application is believed to be in condition for allowance in view of the above comments and amendments. A prompt action to such end is earnestly solicited.

In the event any fees are due in connection with this document, the Commissioner is authorized to charge those fees to Deposit Account No. 50-1063 [MSFTP483US].

Should the Examiner believe a telephone interview would be helpful to expedite favorable prosecution, the Examiner is invited to contact applicants' undersigned representative at the telephone number below.

Respectfully submitted,

AMIN, TUROCY & CALVIN, LLP

/Frank J Schumacher IV/

Frank J Schumacher IV

Reg. No. 61,292

AMIN, TUROCY & CALVIN, LLP
57TH Floor, Key Tower
127 Public Square
Cleveland, Ohio 44114
Telephone (425) 256-8302
Facsimile (216) 696-8731